Tecnológico de Monterrey

Integrated Activity 2

Integrantes:

Gerardo Ulises Sánchez Félix - A01641788

Alan Antonio Ruelas Robles - A01641426

Analysis and Design of Advanced Algorithms

(Gpo 603)

November 28th 2023

For this situation problem we chose four different algorithms in order to solve the four requested tasks corresponding to analyzing the Integrated Activity 2.

Optimal Way to Wire with Optical Fiber (Minimum Spanning Tree):

Problem Statement: Given a graph represented as an adjacency matrix with weighted edges (distances between neighborhoods), find the optimal way to wire neighborhoods with optical fiber so that information can be shared between any two neighborhoods.

Approach (Kruskal's Algorithm): Kruskal's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected, undirected graph. In this context, it's used to find the minimum spanning tree of the neighborhoods, connecting them with the minimum total distance. The edges in this minimum spanning tree represent the optimal way to wire neighborhoods with optical fiber.

In Kruskal's Algorithm, the complexity is typically expressed as O(E log V), where E is the number of edges and V is the number of vertices in the graph.

Here's an explanation of the complexity:

- ➔ The first step in Kruskal's algorithm involves sorting all the edges of the graph by their weights in non-decreasing order.
- ➔ The sorting step has a time complexity of *O(E log E)*, where E is the number of edges.
- ➔ Kruskal's algorithm uses a Union-Find data structure to keep track of the connected components in the graph.
- ➔ The union and find operations in the Union-Find data structure have a time complexity of approximately *O(log V)*, where V is the number of vertices.
- ➔ After sorting, the algorithm iterates through the sorted edges and adds them to the minimum spanning tree if adding the edge does not create a cycle (i.e., the endpoints of the edge belong to different connected components).
- ➔ The total number of iterations in this step is E.
- ➔ In each iteration, a constant time operation (union or find) is performed.

Therefore, the overall time complexity is dominated by the sorting step, resulting in a complexity of *O(E log E)*. However, in a connected graph, the number of edges E is at most V^2, so *O(E log E)* can be expressed as *O(E log V^2)*, and further simplified to *O(2 * E log V)*, and finally to *O(E log V)*.

Shortest Possible Route for Mail Delivery (Repetitive Nearest Neighbor):

Problem Statement: Given the city layout and the need to visit each neighborhood exactly once and return to the origin, find the shortest possible route for mail delivery personnel.

Approach (Repetitive Nearest Neighbor): This is a heuristic approach where the delivery person starts at a neighborhood and repeatedly chooses the nearest unvisited neighborhood until all neighborhoods are visited. The route forms a cycle, and the algorithm continues until all neighborhoods are visited. The algorithm aims to minimize the total distance traveled. The complexity of our algorithm is *O(n^3)* due to the nested call to the nearestNeighbour function. The product of the complexity of the outer loop and the complexity of the nearestNeighbour function:

O(n)×O(nearestNeighbour)=O(n×n^2)=O(n^3)

Maximum Information Flow (Edmonds-Karp Algorithm for Maximum Flow):

Problem Statement: Given the maximum data flow capacities between neighborhoods, find the maximum information flow from the initial node to the final node.

Approach (Edmonds-Karp Algorithm): Edmonds-Karp is an implementation of the Ford-Fulkerson algorithm that uses Breadth-First Search (BFS) to find augmenting paths. In the context of network flow, it's used to find the maximum flow in a network. The neighborhoods and their capacities form a flow network, and the algorithm is used to maximize the information flow from the initial node to the final node.

The time complexity of the Edmonds-Karp algorithm is $O(V * E^2)$, where V is the number of vertices and E is the number of edges in the graph.

Here's a breakdown of the complexity:

- In each iteration of the main loop, the Edmonds-Karp algorithm performs a Breadth-First Search (BFS) to find an augmenting path from the source to the sink.
- BFS has a time complexity of $O(V + E)$ in the worst case.
- The maximum number of iterations in the Edmonds-Karp algorithm is bounded by the maximum flow in the network. In the worst case, the algorithm may iterate $O(E)$ times.
- In each iteration, the algorithm updates the residual capacities of the edges along the augmenting path. This operation takes $O(E)$ time.
- Putting it all together, the overall time complexity is the product of the number of iterations and the complexity of each iteration:

$O(\text{Number of Iterations} \times \text{Complexity of BFS}) = O(E \times (V + E))$

However, in the context of network flow problems, the maximum number of edges E can be at most O(V^2), leading to a simplified time complexity: $O(V \times E^2)$

In summary, the Edmonds-Karp algorithm has a time complexity of O(V * E^2), where V is the number of vertices and E is the number of edges in the graph.

Geographically Closest Exchange (Voronoi Diagram):

Problem Statement: Given the geographic location of exchanges and new service contracts, find the exchange that is geographically closest to a new contract.

Approach (Voronoi Diagram): Voronoi diagrams partition a plane into regions based on the distance to a given set of points. In this case, the set of points represents the locations of exchanges. Each region corresponds to the area closest to a specific exchange. When a new service contract location is given, it belongs to the region of the closest exchange according to the Voronoi diagram.

Reflections:

Gerardo Ulises Sanchez Felix - A01641788

My learning during the elaboration of this project has been an enlightening exploration across diverse dimensions of computer science, providing valuable insights and fostering growth from my standpoint.

Here is what i learned:

- The application of Kruskal's algorithm to determine the optimal wiring for neighborhoods provided a hands-on understanding of graph-related concepts.
- Solving problems related to network optimization, interference considerations, and geographic proximity provided a tangible connection between theoretical concepts and real-world applications. This contextual approach made the learning experience more engaging and applicable.
- The utilization of Voronoi diagrams for exchange selection showcased the practical application of spatial analysis techniques. It was intriguing to witness how such methods can effectively address location-related challenges.

In conclusion, this project provided a multifaceted learning experience, seamlessly integrating theoretical knowledge with practical problem-solving within a collaborative team framework. The skills cultivated are poised to contribute significantly to my ongoing academic and professional journey in computer science.

Alan Antonio Ruelas Robles - A01641426

With all the knowledge, experience and insights that this course provided me, I have realized that there are plenty of algorithms in plenty of areas and subjects where they can be applied in a variety of problems to find exact and approximate solutions, depending on the problem. Although not seeing all of them, I can safely say that we have seen the most important ones in the most important subjects, such as graphs, and some more complex ones like traveling salesman problem and computer graphics algorithms.

This project has helped me to relate how the theory of algorithms seen in class can be applied to real-world problems whose solutions are valuable. Also, it has helped me to improve my problem solving and collaborative skills which are of great importance to my development as a software engineer.